command reference

hp surestore nas 8000 command reference

Edition March 2002
Printed in U.S.A.

Notice

Hewlett-Packard Company makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Format Conventions

WARNING	Identifies a hazard that can cause personal injury
Caution	Identifies a hazard that can cause hardware or software damage
Note	Identifies significant concepts or operating instructions

`Computer font` — used for all text to be typed verbatim: all commands, path names, file names, and directory names also, text displayed on the screen

Italics font — used for variables used in commands

Bold font — used for screen menu options and controls

Trademark Information

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group.

contents

3 Index i

Overview

1

While most of the HP NAS 8000 administrative tasks can be accomplished with the Command View NAS web interface, the HP NAS 8000 also includes a command line interface that allows you to manually enter commands or to run batch commands and scripts using either a serial connection or telnet.

Using the Command Line Interface

To access the NAS 8000 command line interface, you must first log in to the NAS server using a serial connection or telnet.

First, connect to the system.

To connect using a serial connection, use a terminal emulator with the following settings:

- Bits per second: 9600
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow control: None

To connect using telnet, enter the following command on a remote computer:

- `telnet xxx.xxx.xxx.xxx`

where `xxx.xxx.xxx.xxx` is the IP address of the HP NAS 8000 system.

To log in to the system:

- Press **enter** until you see the system name and login prompt.
- Log in as “admin”. No password is required until you set one using the command line interface or the Command View NAS web interface.

```
hp nas8000
NAS OS v1.0.0
localhost login:admin
```

Command Syntax

To execute text commands, execute the commands at the command prompt as shown in the following sections, where:

- % represents the system prompt and is not part of the command to be entered.
- Input parameters (inParam[#]) are entered at the end of the command and are separated by spaces.
- Output parameters (outParam[#]) are displayed on screen or returned to the command script.

Table 1 Sample Command Syntax

`% getNetworkCardList`
`bond0`
`bond1`
`eth0`
	command
	outParam[0]
	outParam[1]
	outParam[2]

`% getNetworkCardIpAddress eth0` `10.1.1.1`	command inParam[0] outParam[0]
`% setNetworkCardIpAddress eth0 10.1.1.2`	command inParam[0] inParam[1]

Scripting

The HP NAS 8000 text command interface is based on Tcl 8.0 (http://www.scriptics.cXom/software/tcltk/8.0.html), which allows for powerful scripting commands such as the following example.

Table 2 Example Command Script

```
% for each i [getNetworkCardList] {
% puts "Network Card Info: $i"
% puts "ip address = [getNetworkCardIpAddress $i]"
% puts "subnet mask = [getNetworkCardSubnetMask $i]"
% }
Network Card Info: bond0
   ip address = 0.0.0.0
   subnet mask = 0.0.0.0
Network Card Info: bond1
   ip address = 0.0.0.0
   subnet mask = 0.0.0.0
Network Card Info: eth0
   ip address = 10.10.0.1
   subnet mask = 255.255.255.0
```

Command Reference

2

Network Card Settings

The HP NAS 8000 has one Network Interface Card (NIC) port on the motherboard and supports two additional slots for NICs. These cards can be either dual-port 10/100 cards or single-port gigabit cards. This support gives the system up to five NIC ports (one on the motherboard and the capacity for a maximum of two dual-port 10/100 NICs).

When you initially set up your HP NAS 8000, you need to configure the primary NIC. After the primary NIC IP address is set, you can specify additional settings for primary and secondary NICs using the HP Command View NAS web interface or by using the following text commands.

getNetworkCardList

Returns list of NIC names and numbers for use in the remaining NIC related commands.

- outParam[0:N] = Network card names

getSystemManagementNetworkCard

Returns the NIC assigned for accessing the Command View NAS web interface.

- outParam[0] = Network card name

setSystemManagementNetworkCard

Sets the NIC assigned for accessing the Command View NAS web interface. The default is the on-board network port (eth0). If you change the port designated as the management port, be sure to note to new IP configuration as this is only way to re-establish communication with your NAS server.

- inParam[0] = Network card name

getNetworkCardBroadcastAddress

Returns the broadcast address for the specified NIC. The broadcast address is the address that can be used to send messages to all machines on the subnet.

- inParam[0] = Network card name
- outParam[0] = Broadcast address

setNetworkCardBroadcastAddress

Sets the Broadcast address for the specified NIC.

- inParam[0] = Network card name
- inParam[1] = Broadcast address

getNetworkCardDhcpEnabled

Identifies whether the specified NIC is DHCP enabled.

When DHCP is enabled, NIC configuration occurs automatically. Depending on your configuration, the DHCP server provides any or all of the following parameters: IP Address, Subnet Mask, Gateway Address, Broadcast Address.

- inParam[0] = Network card name
- outParam[0] = T (Enabled) or F (Disabled)

setNetworkCardDhcpEnabled

Enables or disables DHCP for the specified NIC. If DHCP is disabled, the NIC IP address must be manually defined.

- inParam[0] = Network card name
- inParam[1] = T (Enable) or F (Disable)

getNetworkCardEnabled

Identifies whether the specified NIC is enabled.

- inParam[0] = Network card name
- outParam[0] = T (Enabled) or F (Disabled)

getNetworkCardGatewayAddress

Returns the gateway address for the specified NIC.

- inParam[0] = Network card name
- outParam[0] = Gateway address

setNetworkCardGatewayAddress

Sets the gateway address for the specified NIC.

- inParam[0] = Network card name
- inParam[1] = Gateway address

getNetworkCardInterfaceName

Returns the name for the specified NIC.

- InParam[0] = Network card (NIC)
- OutParam[0] = Name

getNetworkCardIpAddress

Returns the IP address for the specified NIC.

- inParam[0] = Network card
- outParam[0] = IP address

setNetworkCardIpAddress

Sets the IP address for the specified NIC. The IP address and subnet mask for the primary NIC must be defined before you can access the HP Command View NAS web interface from a client system.

- inParam[0] = Network card name
- inParam[1] = IP address

getNetworkCardMacAddress

Returns the MAC address for the specified NIC.

- inParam[0] = Network card name
- outParam[0] = MAC address

getNetworkCardMtu

Returns the Maximum Transmissible Unit (MTU) for the specified gigabit NIC. The MTU is the size of data packets that are sent across the network.

- inParam[0] = Network card name
- outParam[0] = MTU

setNetworkCardMtu

Sets the MTU size for the specified gigabit NIC. The MTU is the size of data packets that are sent across the network.

- inParam[0] = Network card name
- inParam[1] = MTU

getNetworkCardReceiveErrors

Returns the number of errors received by the specified NIC since any NIC configuration changes were made.

- inParam[0] = Network card name
- outParam[0] = Receive Error Count

getNetworkCardReceivePackets

Returns the number of packets received by the specified NIC since any NIC configuration changes were made.

- inParam[0] = Network card name
- outParam[0] = Receive packet count

getNetworkCardSpeed

Returns the speed of the specified NIC.

- inParam[0] = Network card name
- outParam[0] = Speed

getNetworkCardSubnetMask

Returns the subnet mask for the selected NIC.

- inParam[0] = Network card name
- outParam[0] = Subnet mask

setNetworkCardSubnetMask

Sets the subnet mask for the selected NIC.

- inParam[0] = Network card name
- inParam[1] = Subnet mask

getNetworkCardTransmitBytes

Returns the number of bytes transmitted by the specified NIC since any NIC configuration changes were made.

- inParam[0] = Network card name
- outParam[0] = Transmit byte count

getNetworkCardTransmitCollisions

Returns the number of transmit collisions experienced by the specified NIC since any NIC configuration changes were made.

- inParam[0] = Network card name
- outParam[0] = Transmit collision count

getNetworkCardTransmitPackets

Returns the number of packets transmitted by the specified NIC since any NIC configuration changes were made.

- inParam[0] = Network card name
- outParam[0] = Transmit packet count

Network Card Bonding

When you configure NIC ports, you have the option of enabling bonding, a failover mechanism that automatically switches a specific NIC port to a standby network upon the failure or abnormal termination of the currently active system. By default, the lowest-numbered NIC provides the addresses of the bond channel, but you can change the addresses manually.

Note Enabling DHCP disables NIC bonding and enabling NIC bonding disables DHCP. Enabling DHCP or NIC bonding disables manually configuring addresses.The following commands are used to configure NIC bonding.

getNetworkCardBondMasterEnabled

Identifies whether the specified NIC is a bond master.

- inParam[0] = Network card name
- outParam[0] = T (Enabled) or F (Disabled)

getNetworkCardBondSlaveEnabled

Identifies whether the specified NIC is a bond slave.

- inParam[0] = Network card name
- outParam[0] = T (Enabled) or F (Disabled)

bondEnslaveNetworkCard

Sets the specified NIC to be a bond slave of the specified master NIC.

- inParam[0] = Network card name
- inParam[1] = Bond master card handle

bondReleaseNetworkCard

Releases the specified NIC from bonds.

- inParam[0] = Network card name

getNetworkCardBondMonitorInterval

Returns the bond monitoring interval (in milliseconds) for the selected NIC.

- inParam[0] = Network card name
- outParam[0] = Bond monitor interval

setNetworkCardBondMonitorInterval

Sets the bond monitoring interval (in milliseconds) for the selected NIC.

- inParam[0] = Network card name
- inParam[1] = Bond monitor interval

getNetworkCardBondUpDelay

Returns the time (in milliseconds) for the selected NIC to delay after bondEnslave before bringing the bond link up.

- inParam[0] = Network card name
- outParam[0] = Bond up delay

setNetworkCardBondUpDelay

Sets the time (in milliseconds) for the specified NIC to delay after bondEnslave before bringing the bond link up.

- inParam[0] = Network card name
- inParam[1] = Bond up delay

getNetworkCardBondDownDelay

Returns the time (in milliseconds) for the specified NIC to delay after bondRelease before taking the bond link down.

- inParam[0] = Network card name
- outParam[0] = Bond down delay

setNetworkCardBondDownDelay

Sets the time (in milliseconds) for the specified NIC to delay after bondRelease before taking the bond link down.

- inParam[0] = Network card name
- inParam[1] = Bond down delay

getNetworkCardBondMasterInterfaceName

Returns the interface name assigned to the specified bond master NIC.

- inParam[0] = Network card name
- outParam[0] = Bond master interface name

getNetworkCardBondActiveSlaveList

Returns a list of network cards that are both bond slaves and active (that is, their bond master has failed).

- inParam[0] = Bond master NIC handle
- outParam[0:N] = Slave network card names

getNetworkCardBondSlaveList

Returns a list of network cards that are bond slaves.

- inParam[0] = Bond master NIC handle
- outParam[0:N] = Slave network card names

getNetworkCardBondMasterList

Returns a list of network cards that are bond masters.

- outParam[0:N] = Network card names

Network Settings

Domain Name Service (DNS) Settings

Domain Name Servers convert system names that people can remember (such as nas8000.fc.hp.com) to IP addresses (such as 123.45.67.89) that are used by packet-routing software.

The following commands can be used to enter and edit DNS information.

getNetworkDnsDomainName

Returns the DNS domain name.

- outParam[0] = DNS domain name

setNetworkDnsDomainName

Sets the DNS domain name. The HP NAS 8000 can belong to only one domain.

- inParam[0] = DNS domain name

getNetworkDnsAddressList

Returns the list of DNS addresses associate with the HP NAS 8000.

- outParam[0:N] = DNS address list

setNetworkDnsAddressList

Up to three DNS Server Addresses can be specified. These should be entered in the appropriate search order. In other words, enter the IP address of the Primary DNS first followed by the IP address of the secondary DNS and so on until all of your Domain Name Servers have been identified.

- inParam[0:N] = DNS address list

Values must be separated by spaces, for example:

- %setNetworkDnsAddressList 10.1.1.1 10.1.1.2 10.1.1.3

Network Host Settings

getNetworkHostName

Returns the network host name.

- outParam[0] = Host name

setNetworkHostName

Sets the network host name. If the full host name is nas8000.fc.hp.com, then nas8000 is the host name and fc.hp.com is the domain name.

- inParam[1] = Host name

Network NIS Settings

The HP NAS 8000 supports Network Information System (NIS). NIS maintains a central database of names and locations of resources on a network. NIS was formerly known as Yellow Pages. To enable or disable NIS, use the following commands.

getNetworkNisDomainName

Returns the NIS domain name.

- outParam[0] = NIS domain name

setNetworkNisDomainName

Sets the NIS domain name for the HP NAS 8000.

- inParam[0] = NIS domain name

getNetworkNisEnabled

Returns whether the HP NAS 8000 is NIS enabled.

- outParam[0] = T (Enabled) or F (Disabled)

setNetworkNisEnabled

Enables or disables NIS for the HP NAS 8000.

- inParam[0] = T (Enable) or F (Disable)

getNetworkNisCurrentServerName

Returns the name of the NIS server the HP NAS 8000 system is currently connected to.

- outParam[0] = Current NIS server name

getNetworkNisServerName

Returns the name of the NIS server the HP NAS 8000 will attempt to connect to at boot.

- outParam[0] = NIS server name

setNetworkNisServerName

Sets the NIS server the HP NAS 8000 system will attempt to connect to at boot.

- inParam[0] = NIS server name

Network SNMP Settings

If you are using a Simple Network Management Protocol (SNMP) tool, you can define the names or IP addresses of hosts to receive notification in case of an event.

This information is optional and does not affect the functionality of the device. In the event of a hardware failure or system alert, messages are sent via SNMP traps, email (SMTP), and logged in the system log.

getNetworkSnmpTrapDestVersion

Returns the version of SNMP protocol to use when sending a trap to this Trap Destination (currently only v1 and v2c are supported; default: v1).

- inParam[0] = Destination number
- outParam[0] = Value

getNetworkSnmpCommunityString

Returns the SNMP community name for this device (default: public).

- outParam[0] = SNMP community name

setNetworkSnmpCommunityString

Sets the SNMP community name for this device (default: public).

- inParam[0] = SNMP community name

getNetworkSnmpTrapDestVersion

Returns the version of SNMP protocol to use when sending a trap to this Trap Destination (currently only v1 and v2c are supported; default: v1).

- inParam[0] = Destination number
- outParam[0] = Value

setNetworkSnmpTrapDestVersion

Sets the version of SNMP protocol to use when sending a trap to this Trap Destination (currently only v1 and v2c are supported; default: v1).

- inParam[0] = Destination number
- inParam[1] = Value

getNetworkSnmpTrapDestCommunity

Returns the SNMP community name for this Trap Destination (default: public).

- inParam[0] = Destination number
- outParam[0] = Value

setNetworkSnmpTrapDestCommunity

Sets the SNMP community name for this Trap Destination (default: public).

- inParam[0] = Destination number
- inParam[1] = Value

getNetworkSnmpTrapDestDomain

Returns the Domain for this Trap Destination (currently only “UDP” is supported).

- inParam[0] = Destination number
- outParam[0] = Value

setNetworkSnmpTrapDestDomain

Sets the Domain for this Trap Destination (currently only “UDP” is supported).

- inParam[0] = Destination number
- inParam[1] = Value

getNetworkSnmpTrapDestAddress

Returns the address for this Trap Destination.

- inParam[0] = Destination number
- outParam[0] = Value

setNetworkSnmpTrapDestAddress

Sets the address for this Trap Destination. May be numeric (1.2.3.4) or DNS hostname (snmp.hp.com).

- inParam[0] = Destination number
- inParam[1] = Value

getNetworkSnmpTrapDestPort

Returns the port set for the SNMP trap (default: 162).

- inParam[0] = Destination number
- outParam[0] = Port

setNetworkSnmpTrapDestPort

Sets the port number for the SNMP trap (default: 162).

- inParam[0] = Destination number
- inParam[1] = Port

setNetworkSnmpTrapDestStatus

Sets the Status for this Trap Destination. See www.hp.com/support/nas8000 for information on trap status.

- inParam[0] = Destination number
- inParam[1] = Value

getNetworkSnmpTrapDestStatus

Returns the Status for this Trap Destination. See www.hp.com/support/nas8000 for information on trap status.

- inParam[0] = Destination number
- outParam[0] = Value

getNetworkSnmpTrapDestList

Returns the complete list of Trap Destination Addresses.

- outParam[0:N] = List of trap destination hostnames or IP addresses

setNetworkSnmpTrapDestList

Sets the complete list of Trap Destination Addresses. Deletes all current Trap Destinations, and creates new ones based on this list of addresses. (Status is set to "active"; all other parameters are not created, and will return defaults).

- inParam[0:N] = List of trap destination hostnames or IP addresses

getNetworkSnmpActiveTrapDestList

Returns the complete list of "active" Trap Destinations.

- outParam[0:N] = List of trap destination hostnames or IP addresses

getNetworkSnmpCommunityString

Returns the SNMP community name for this device (default: public).

- outParam[0] = SNMP community name

setNetworkSnmpCommunityString

Sets the SNMP community name for this device (default: public).

- inParam[0] = SNMP community name

Network Client Settings

These commands provide information about the network clients connected to the HP NAS 8000 system.

getNetworkClientInfoList

Returns a list of network clients currently connected to the HP NAS 8000 storage.

- outParam[0:N] = Network client IDs

getNetworkClientInfoProtocol

Returns the protocol used by the specified network client.

- inParam[0] = Network client ID
- outParam[0] = Network protocol (telnet, NFS, Windows®, etc.)

getNetworkClientInfoName

Returns the network name of the specified client.

- inParam[0] = Network client ID
- outParam[0] = Network name (xxx.xxx.xxx.xxx or DNS name)

getNetworkClientInfoPath

Returns the path used by the specified client (that is, what part of the storage are they using).

- inParam[0] = Network client ID
- outParam[0] = Path used by client (if applicable)

getNetworkClientInfoVolume

Returns the name of the file volume being used by the specified client.

- inParam[0] = Network client ID
- outParam[0] = Volume used by client (if applicable)

getNetworkClientInfoVolumeDir

Returns the name of the sub-directory being used by the specified client.

- inParam[0] = Network client ID
- outParam[0] = Sub-directory used by client within volume used by client (if applicable)

getNetworkClientInfoConnectTime

Returns the time the specified client connected to the HP NAS 8000 storage.

- inParam[0] = Network client ID
- inParam[1] = Optional format string (default is "%c")
- outParam[0] = Date & Time according to format (default is "%m/%d/%Y%T")

For display, you should always select one of the "locale" safe formats:

- %c - TIME & DATE for locale
- %x - DATE for locale
- %X - TIME for locale
- %r - Time am/pm (exact format may change with locale)
- %R - %H:%M
- %T - %H:%M:%S (24 hour)
- %D - %m/%d/%y - (United States specific format)

getNetworkClientInfoComment

Returns the comment associated with the specified network client.

- inParam[0] = Network client ID
- outParam[0] = Extra information about network client (if applicable)

Misc. Network Settings

getNetworkConfigured

Identifies whether or not the network is configured.

- outParam[0] = T (configured) or F (not configured)

Storage Settings

Storage settings allow you to control Direct Access Storage Devices (DASDs) connected to the HP NAS 8000 solution. DASDs include disk drives and storage arrays.

getStorageDasdList

Returns a list of Direct Access Storage Devices attached to the HP NAS 8000 system.

- outParam[0:N] = DASD name list

getStorageDasdType

Returns the DASD type of the specified device.

- inParam[0] = DASD name
- outParam[0] = DASD type

Where DASD type can be one of:

- Va71xx = HP VA71xx storage array
- Va74xx = HP VA74xx storage array
- GenericScsi = Any other SCSI disk or array

getStorageDasdSerialNumber

Returns the serial number of the specified storage device.

- inParam[0] = DASD name
- outParam[0] = DASD serial number

getStorageDasdVendor

Returns the vendor name of the specified DASD (as returned by SCSI inquiry).

- inParam[0] = DASD name
- outParam[0] = DASD vendor

getStorageDasdProduct

Returns the product IS of the specified DASD (as returned by SCSI inquiry).

- inParam[0] = DASD name
- outParam[0] = DASD product

getStorageDasdSerialNumber

Returns the serial number of the specified DASD.

- inParam[0] = DASD name
- outParam[0] = DASD serial number

getStorageDasdDeviceFile

Returns the name of the device file used to access the specified DASD.

- inParam[0] = DASD name
- outParam[0] = DASD device file

getStorageDasdScsiList

Returns a list of the storage devices attached to or allocated for use with the NAS 8000.

- outParam[0:N] = SCSI DASD name

getStorageDasdScsiGenericDeviceFile

Returns the absolute path to the SCSI generic device file for a DASD.

- inParam[0] = DASD name
- outParam[0] = DASD generic device file

getStorageDasdScsiHost

Returns the SCSI driver host number for a DASD.

- inParam[0] = SCSI DASD name
- outParam[0] = Host number

getStorageDasdScsiChannel

Returns the SCSI HBA channel number used to access the SCSI DASD.

- inParam[0] = DASD name
- outParam[0] = DASD channel number

getStorageDasdScsiTargetId

Returns the SCSI target ID number used to access the SCSI DASD.

- inParam[0] = SCSI DASD name
- outParam[0] = SCSI DASD target ID

getStorageDasdScsiMaxNumberLuns

Returns the maximum number of LUNs supported by the system.

- outParam[0] = Maximum number of user data LUNs

getStorageDasdScsiVa71xxList

Returns a list of the storage arrays connected to the HP NAS 8000 system.

- outParam[0:N] = SCSI DASD Va71xx name list

getStorageDasdScsiVa74xxList

Returns a list of VA74xx disk arrays attached to the system.

- outParam[0:N] = SCSI DASD Va7xxx name list

getStorageDasdScsiVA7xxxList

Returns a list of disk arrays attached to the system.

- outParam[0:N] = SCSI DASD Va7xxx name list

getStorageDasdScsiVa7xxxManagementSoftwareRunning

Informs whether Array Management Software is running.

- outParam[0] = T (running) or F (not running)

getStorageDasdScsiVa7xxxAlias

Returns the alias for the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0] = Alias

setStorageDasdScsiVa7xxxAlias

Sets the alias for the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Alias

getStorageDasdScsiVa7xxxUniqueName

Returns the unique name of the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0] = Unique name

getStorageDasdScsiVa7xxxHealth

Returns health status of the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0] = Health

getStorageDasdScsiVa7xxxRebuilding

Informs whether the specified storage array is in the process of rebuilding data.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0] = T (rebuilding) or F (not rebuilding)

getStorageDasdScsiVa7xxxRebuildPercentComplete

When a storage array is rebuilding data, Returns the percent complete.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0] = Rebuilding percentage (0-99)

getStorageDasdScsiVa7xxxCapacities

Returns the capacity and other configurations of the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Total capacity
- outParam[1] = Unallocated capacity
- outParam[2] = Allocated capacity
- outParam[3] = Unincluded capacity
- outParam[4] = Included capacity
- outParam[5] = Redundant capacity
- outParam[6] = Active spare capacity

getStorageDasdScsiVa7xxxRedundancygroupMinimumNumber

Returns the lowest numbered redundancy group of a VA7xxx array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Redundancy group
- outParam[0] = Minimum redundancy group

getStorageDasdScsiVa7xxxRedundancygroupMaximumNumber

Returns the highest numbered redundancy group of a VA7xxx array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Redundancy group
- outParam[0] = Maximum redundancy group

getStorageDasdScsiVa7xxxTotalCapacity

Returns the total raw capacity of the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Device capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxUnallocatedCapacity

Returns the unallocated capacity – that is, space not included in volume groups - for the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Device capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxAllocatedCapacity

Returns the allocated capacity – that is, space included in volume groups – for the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Device capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxActiveSpareCapacity

Returns the capacity of the active spare on the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Device capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxUnincludedCapacity

Displays the unincluded capacity – that is, space not included in LUNs – of the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Device capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxIncludedCapacity

Gets the included capacity of a redundancy group on a VA7xxx storage array. Included capacity is the sum of the capacities of all disk drives that are in use.

- inParam[0] = VA7xxx DASD name
- inParam[1] = Redundancy group (* for all redundancy groups).
- inParam[2] = Units (B,MB,GB)
- outParam[0] = Included capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxRedundantCapacity

Returns the redundant capacity – that is, the space used for data redundancy for the current RAID setting – on the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Device capacity (MB - unless overridden by inParam[1])

scanStorageDasdScsiVa7xxx

Updates the list of arrays connected to the HP NAS 8000 system. Used when devices are added or removed.

- inParam[0] = SCSI VA7xxx DASD name

shutdownStorageDasdScsiVa7xxx

Shuts down the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name

restartStorageDasdScsiVa7xxx

Restarts the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name

getStorageDasdScsiVa7xxxLunList

Returns a list of LUNs on the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0:N] = LUN number list

getStorageDasdScsiVa7xxxLunMinimumCapacity

Returns the minimum size that can be used to create a LUN.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = LUN minimum capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxLunCapacityIncrement

LUN sizes must be defined using this increment. Default is 1MB.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = LUN capacity increment (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxLunMinimumNumber

Returns the minimum allowed LUN number. Currently 1.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0] = Minimum LUN number

getStorageDasdScsiVa7xxxLunMaximumNumber

Returns the maximum allowed LUN number. Currently 127.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0] = Maximum LUN number

addStorageDasdScsiVa7xxxLun

Adds a LUN of the specified size to the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = LUN number
- inParam[2] = Device capacity (MB)

removeStorageDasdScsiVa7xxxLun

Deletes the specified LUN from the specified storage array. All data will be lost.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = LUN number

getStorageDasdScsiVa7xxxLunCapacity

Returns the capacity of the specified LUN on the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = LUN number
- inParam[2] = Units (B,MB,GB)
- outParam[0] = Device capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiVa7xxxLun

Returns the name of the specified LUN on the specified storage array.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = LUN number
- outParam[0] = SCSI LUN name

getStorageDasdScsiVa7xxxLunRedundancygroup

Returns the redundancy group of a VA7xxx array LUN. Note the difference between this command and the very similar **getStorageDasdScsiLunRedundancygroup** command.

- inParam[0] = SCSI VA7xxx DASD name
- inParam[1] = LUN number
- outParam[0] = Redundancy group

getStorageDasdScsiLunRedundancygroup

Returns the redundancy group number that this DASD LUN is on. This command should only be used on LUNs that are on an HP VA7xxx storage array.

- inParam[0] = SCSI LUN name
- outParam[0] = Redundancy group number

getStorageDasdScsiVa7xxxPerformanceLogList

Returns a list of paths to performance log files for a VA7xxx storage array. Returned paths are relative to the web-server document root directory.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0:N] = Downloadable performance log file paths

getStorageDasdScsiVa7xxxDeviceLogList

Returns a list of paths to diagnostic log files for a VA7xxx storage array. Returned paths are relative to the web-server document root directory.

- inParam[0] = SCSI VA7xxx DASD name
- outParam[0:N] = Downloadable diagnostic log file names

getStorageDasdScsiPerDriveLunList

Returns a list of LUNs available on the specified array or device.

- inParam[0] = SCSI DASD name
- outParam[0:N] = SCSI DASD LUN list

getStorageDasdScsiLunList

Returns a list of LUNs that are being used in volume groups.

- outParam[0:N] = SCSI LUN name list

getStorageDasdScsiInternalLunList

Returns a list of SCSI DASD LUNs that are on internally-mounted DASDs. Internal DASD LUNs are used for system functions. They are never used to store user data.

- outParam[0:N] = SCSI LUN name list

getStorageDasdScsiExternalLunList

Returns a list of SCSI LUNs that are part of externally-attached (usually through a Fibre-channel connection) DASDs. This command currently returns the same list at the **getStorageDasdScsiLunList** command.

- outParam[0:N] = SCSI LUN name list

getStorageDasdScsiAvailableLunList

Returns a list of LUNs that are available for use in volume groups.

- outParam[0:N] = Available LUN number list

getStorageDasdScsiLunBlockDeviceFile

Returns the block (SCSI disk driver) device file used to access a SCSI LUN.

- inParam[0] = SCSI LUN name
- outParam[0] = Block device file

getStorageDasdScsiLunRawDeviceFile

Returns the raw device file used to access a SCSI LUN. Raw device files are used to bypass the system cache when writing or reading data to or from the device.

- inParam[0] = SCSI LUN name
- outParam[0] = Raw device file

getStorageDasdScsiLunGenericDeviceFile

Returns the generic device file used to access a SCSI LUN. Generic device files are most commonly used to send raw SCSI commands to devices.

- inParam[0] = SCSI LUN name
- outParam[0] = Generic device file

getStorageDasdScsiLunRedundancygroup

Returns the redundancy group number that this DASD LUN is on. This command should only be used on LUNs that are on an HP VA7xxx storage array.

- inParam[0] = SCSI LUN name
- outParam[0] = Redundancy group number

getStorageDasdScsiLunNumber

Returns the number of the specified LUN.

- inParam[0] = SCSI LUN name
- outParam[0] = LUN number

getStorageDasdScsiLunCapacity

Returns the capacity of the specified LUN.

- inParam[0] = SCSI LUN name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = LUN capacity (MB - unless overridden by inParam[1])

getStorageDasdScsiLunInUse

Informs whether the specified LUN is in use, that is, whether it is part of a volume group.

- inParam[0] = SCSI LUN name
- outParam[0] = T (in use) or F (Not in use)

getStorageDasdScsiLunDasd

Returns the name of the specified LUN.

- inParam[0] = SCSI LUN name
- outParam[0:N] = DASD name

Volume Group Settings

A volume group is the aggregation of one or more LUNs. Volume groups combine the space from LUNs and make the space accessible to the file system for creating file volumes and directories, which can then be made accessible to users. The following text commands are used to create and manage volume groups.

getStorageVolumegroupSystemList

Returns a list of existing system-data volume group names.

- outParam[0:N] = Volume group name list

getStorageVolumegroupUserList

Returns a list of existing user data-volume group names. This command is exactly equivalent to the **getStorageVolumegroupList** command.

- outParam[0:N] = Volume group name list

getStorageVolumegroupExists

Answers: Is there a volume group with name?

- InParam[0] = Volume group name
- outParam[0] = T (true) or F (false)

getStorageVolumegroupActivated

Identifies whether or not the specified volume group has been activated.

- inParam[0] = Volume group name
- outParam[0] = T (true) or F (false)

addStorageVolumegroupUser

Creates a new user-data volume group from a list of DASD LUNs. This command is exactly equivalent to the **addStorageVolumegroup** command.

- inParam[0] = Volume group name
- inParam[1:N] = DASD LUN names

getStorageVolumegroupList

Returns a list of volume groups that have been created on the HP NAS 8000 system.

- outParam[0:N] = Volume group name list

addStorageVolumegroup

Creates a new volume group containing the space from the specified LUNs.

- inParam[0] = Volume group name
- inParam[1:N] = LUN names

addStorageVolumegroupLuns

Adds the specified LUNs to an existing volume group.

- inParam[0] = Volume group name
- inParam[1:N] = LUN names

deleteStorageVolumegroup

Deletes the specified volume group. All data will be lost.

- inParam[0] = Volume group name

deleteStorageVolumegroupLuns

Removes the specified LUNs from an existing volume group. All data will be lost.

- inParam[0] = Volume group name
- inParam[1:N] = LUN names

setStorageVolumegroupName

Renames an existing volume group.

- inParam[0] = Existing volume group name
- inParam[1] = New volume group name

getStorageVolumegroupAvailableCapacity

Returns the available capacity – that is the amount of free space – on the specified volume group.

- inParam[0] = Volume group name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Available Capacity (MB - unless overridden by inParam[1])

getStorageVolumegroupTotalCapacity

Returns the total raw capacity of the specified volume group.

- inParam[0] = Volume group name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Total Capacity (MB - unless overridden by inParam[1])

getStorageVolumegroupVolumeMinimumCapacity

Returns the minimum size that can be used to create a file volume or snapshot on the specified volume group.

- inParam[0] = Volume group name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Minimum Volume Capacity (MB - unless overridden by inParam[1])

getStorageVolumegroupVolumeMaximumCapacity

Returns the maximum size that can be used to create a file volume or snapshot on the specified volume group.

- inParam[0] = Volume group name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Maximum Volume Capacity (MB - unless overridden by inParam[1])

getStorageVolumegroupVolumeCapacityIncrement

Returns the size increment that can be used to create a file volume or snapshot on the specified volume group. Currently 32MB.

- inParam[0] = Volume group name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Volume Capacity Increment (MB - unless overridden by inParam[1])

getStorageVolumegroupLunList

Returns a list of the LUNs contained in the specified volume group.

- inParam[0] = Volume group name
- outParam[0:N] = LUN handle list

getStorageVolumegroupVolumeList

Returns a list of the file volumes that exist in the specified volume group.

- inParam[0] = Volume group name
- outParam[0:N] = Volume name list

Volume Settings

A volume group is divided into one or more file volumes. File volumes are the basic unit of logical storage for a file system on the HP NAS 8000. File volumes can be further subdivided into individual directories. The following commands are used to create and manage file volumes.

getStorageVolumeType

Returns the type of the specified volume group: file volume or snapshot.

- InParam[0] = Volume name
- outParam[0] = Volume Type

getStorageVolumeList

Returns a list of file volumes that exist on the storage arrays in the HP NAS 8000 solution.

- outParam[0:N] = Volume name list

setStorageVolumeSize

Sets the size of the specified file volume.

- inParam[0] = Volume name
- inParam[1] = Size (MB)

deleteStorageVolume

Deletes the specified file volume. All data will be lost.

- inParam[0] = Volume name

setStorageVolumeName

Renames an existing file volume.

- inParam[0] = Existing Volume name
- inParam[1] = New Volume name

getStorageVolumeExists

Identifies whether or not the specified file volume exists. Answers: Is there a volume with name?

- InParam[0] = Volume name
- outParam[0] = T (true) or F (false)

getStorageVolumeNumber

Returns the number of the specified file volume.

- inParam[0] = Volume name
- outParam[0] = Volume number

getStorageVolumeVolumegroup

Returns the name of the volume group that contains the specified file volume.

- inParam[0] = Volume name
- outParam[0] = Volume group name

getStorageVolumeAvailableCapacity

Returns the amount of available, or free, space in the specified file volume.

- inParam[0] = Volume name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Available Capacity (MB - unless overridden by inParam[1])

getStorageVolumeTotalCapacity

Returns the total raw space in the specified file volume.

- inParam[0] = Volume name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Total Capacity (MB - unless overridden by inParam[1])

getStorageVolumeAtMaximumNumberVolumes

Identifies whether the maximum number of file volumes exist on the HP NAS 8000 solution.

- outParam[0] = T (max reached) or F (below max)

getStorageVolumeMountPath

Returns the mount path, if any, that is defined for UNIX® sharing of the specified file volume.

- inParam[0] = Volume name
- outParam[0] = Volume mount path

getStorageVolumeDeviceFile

Returns the name of the device file that is used to access the specified file volume.

- inParam[0] = Volume name
- outParam[0] = Volume device file

getStorageVolumeRootPath

Returns the path to the location where user data is stored on the specified file volume.

- inParam[0] = Volume name
- outParam[0] = Volume file system root path

getStorageVolumeVolumeMinimumCapacity

Returns the minimum capacity of the specified storage volume.

- inParam[0] = Volume name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Minimum volume capacity (MB - unless overridden by inParam[1])

getStorageVolumeVolumeMaximumCapacity

Returns the maximum capacity for the specified storage volume.

- inParam[0] = Volume name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Minimum volume capacity (MB - unless overridden by inParam[1])

getStorageVolumeVolumeCapacityIncrement

Returns the capacity increment (in bytes, megabytes, and gigabytes) used to create or resize file and snapshot volumes, i.e., when creating volumes, the capacity modulo increment must be equal to 0.

- inParam[0] = Volume name
- inParam[1] = Units (B,MB,GB)
- outParam[0] = Minimum volume capacity (MB - unless overridden by inParam[1])

getStorageVolumeFileList

Returns a list of file volumes that exist on the HP NAS 8000 system.

- outParam[0:N] = Volume name list

getStorageVolumeFileSystemList

Returns a list of file volumes.

- outParam[0:N] = Volume name list

getStorageVolumeFileUserList

Returns a list of existing user-data file volume names.

- outParam[0:N] = Volume name list

getStorageVolumeFileType

Returns the file volume type.

- inParam[0] = File volume name
- outParam[0] = File volume type

Where type can be one of:

- Unknown
- System
- User

addStorageVolumeFile

Creates a new file volume in the specified volume group.

- inParam[0] = Volume name
- inParam[1] = Volume group name
- inParam[2] = Size (MB)

Snapshot Settings

A snapshot is a read-only picture of a file volume at a specific point in time. When you create a file volume, the snapshot of that file volume is of zero length. However, as you modify the file volume, the snapshot tracks changes between the original file volume and the modified file volume. If an error occurs and you want to revert to the previous version, you can use the snapshot data and the unmodified parts of the original file volume to quickly and easily construct the file volume.

When you set up a snapshot, consider how quickly the file's data will change, and how often you will delete snapshots and start over. You have the option to size snapshots relatively small and let them autogrow. As they reach the limit that you establish, their size can increase by approximately 10 percent to accommodate the changes. This flexibility lets you set the snapshot and not worry about a specific size.

After you specify a size, you must define an expiration date. When the snapshot expires, the system automatically deletes it. Consider your overall backup strategy in light of the snapshot expiration date. For example, you may want to take a snapshot of your data for a specified amount of time, then when you are certain that you have a backup of your system, delete the snapshot and begin the snapshot process again.

The system treats snapshots as part of a regular file volume. The following commands are used to create and manage snapshots.

getStorageVolumeFileVolumeSnapshotList

Returns a list of snapshots that exist for the specified file volume.

- inParam[0] = File volume name
- outParam[0:N] = Snapshot name list

getStorageVolumeFileVolumeSnapshotCount

Returns a count of how many snapshots have been taken of the specified file volume.

- inParam[0] = File volume name
- outParam[0] = Snapshot count

getStorageVolumeSnapshotList

Returns a list of the snapshots that exist on the HP NAS 8000 system.

- outParam[0:N] = Snapshot name list

addStorageVolumeSnapshot

Creates a new snapshot of the specified file volume.

Note Adequate space must exist in the volume group that contains the file volume.

- inParam[0] = Snapshot name
- inParam[1] = Volume name
- inParam[2] = Size (MB)

getStorageVolumeSnapshotVolumeFile

Returns the name of the file volume associated with the specified snapshot.

- inParam[0] = Snapshot name
- outParam[0] = Volume name

getStorageVolumeSnapshotExpiration

Returns the expiration date and time for the specified snapshot.

- inParam[0] = Snapshot name
- inParam[1] = Optional date format string (defaults to "%m/%d/%Y")
- inParam[2] = Optional time format string (defaults to "%T")

For display, you should always select one of the "locale" safe formats:

- %c - TIME & DATE for locale
- %x - DATE for locale
- %X - TIME for locale
- %r - Time am/pm (exact format may change with locale)
- %R - %H:%M
- %T - %H:%M:%S (24 hour)
- %D - %m/%d/%y - (United States specific format)

setStorageVolumeSnapshotExpiration

Sets the date and time that the specified snapshot will expire.

- inParam[0] = Snapshot name
- inParam[1] = Date (mm/dd/yyyy) or "NOW"
- inParam[2] = Time (24hh:mm:ss) or "NOW"

OR, sets the amount of time the specified snapshot will exist before expiring.

- inParam[1] = [-hours | -days | -weeks | -months | -years]
- inParam[2] = Count

getStorageVolumeSnapshotAutogrowEnabled

Identifies whether the specified snapshot will automatically grow when it nears full capacity.

- inParam[0] = Volume name
- outParam[0] = T (Enabled) or F (Disabled)

setStorageVolumeSnapshotAutogrowEnabled

Enables or disables the specified snapshot autogrow.

- inParam[0] = Volume name
- inParam[0] = T (Enabled) or F (Disabled)

Storage Quota Settings

Quotas are set on a per file volume basis. You can have a quota larger than the available space on the file volume.

The following commands are used to manage storage quotas. For information on setting quotas, see User and Group Quota Settings on page User and Group Quota Settings.

getStorageQuotaEnabled

Identifies whether quotas are enabled for the specified file volume.

- inParam[0] = Volume name
- outParam[0] = T (Enabled) or F (Disabled)

setStorageQuotaEnabled

Enables or disables quotas for the specified file volume.

- inParam[0] = Volume name
- inParam[0] = T (Enabled) or F (Disabled)

Share Settings

Before network users can access the NAS server, you must give them permission. This is a security concern. Each platform grants permission differently:

- **Windows**: Shares are permissions that let you control Microsoft Windows users' access to data. You can create shares for any directory within a file volume, including the root. Once a share is created, users may attach to the share via the Network Neighborhood in Windows and store and retrieve files and directories. If you are operating under share-level security, you can limit access to shares by creating read-only or read/write passwords. Windows-specific commands are covered in SMB settings.
- **UNIX**: You create an export so that users can mount that volume/directory on their systems. However, you must first specify the access mode. If you specify a read-only or read/write access mode, users can use the `mount` command to access the volume from a UNIX workstation. This restriction is only for general access to the system. User-level restrictions also apply to all of the files and directories on the volume. Unix-specific commands are covered in NFS settings.

Shares and exports let you control which data clients can access. Host access allows you to control which client machines are allowed access to the HP NAS 8000, regardless of the user.

getStorageShareList

Returns a list of shares and exports that exist for the specified file volume, or for all file volumes if none is specified.

- inParam[0] = Volume name (if "*", then do all shares for all volumes)
- inParam[1] = Optional - if present, anything (true, 1, yes) means output Volume & Share
- outParam[0:N] = Share path list - If inParam[1] exits, format is 'VolumeName:_*_:SharePath'

getStorageShareCount

Returns the number of shares on the specified file volume.

- inParam[0] = File volume name
- outParam[0] = Share count

addStorageShare

Creates a new share/export for the specified file volume.

- inParam[0] = Volume name
- inParam[1] = Share path

removeStorageShare

Removes the specified share/export from the specified file volume.

- inParam[0] = Volume name
- inParam[1] = Share path

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path

getStorageSharePath

Returns the path to the specified share/export.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0:N] = Host allow list for this share

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = Path for the share.

getStorageShareHostsAllowList

Returns a list of the hosts that will be given access to the storage on the HP NAS 8000 system.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0:N] = Host allow list for this share

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0:N] = Host allow list for this share

setStorageShareHostsAllowList

Sets the hosts that will be given access to the storage on the HP NAS 8000 system.

- inParam[0] = Volume name
- inParam[1] = Share path
- inParam[2:N] = Host allow list for this share

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- inParam[1:N] = Host allow list for this share

SMB Settings

SMB/CIFS, the Windows protocol for sharing files, lets client applications read and write to files. CIFS is a proposed standard protocol that lets programs request files and services on remote computers on the internet. CIFS uses the client/server programming model. A client program makes a request of a server program (usually running on another computer) for access to a file or to pass a message to a program that runs on the server computer. The server takes the requested action and returns a response.

The following settings apply to Windows SMB file sharing.

getStorageShareSmbEnabled

Identifies whether Windows SMB sharing is enabled for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0] = T (Enabled) or F (Disabled)

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = T (Enabled) or F (Disabled)

setStorageShareSmbEnabled

Enables or disables Windows SMB sharing for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- inParam[2] = T (Enable) or F (Disable)
- inParam[3] = SMB name (Only if enabling)

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- inParam[1] = T (Enable) or F (Disable)
- inParam[2] = SMB name (Only if enabling)

getStorageShareSmbName

Returns the Windows SMB name of the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0] = SMB name

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = SMB name

getStorageShareSmbComment

Returns the Windows SMB comment associated with the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0] = Comment

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = Comment

setStorageShareSmbComment

Sets the Windows SMB comment associated with the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- inParam[2] = Comment

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- inParam[1] = Comment

getStorageShareSmbRwPassword

Returns the Windows SMB read/write password for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0] = Read/Write password

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = Read/write password

setStorageShareSmbRwPassword

Sets the Windows SMB read/write password for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- inParam[2] = Read/Write password

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- inParam[1] = Read/write password

getStorageShareSmbRoPassword

Returns the Windows SMB read-only password for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0] = Read-only password

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = Read-only Password

setStorageShareSmbRoPassword

Sets the Windows SMB read-only password for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- inParam[2] = Read-only password

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- inParam[1] = Read-only Password

getNetworkSmbComment

Returns the comment field associated with the network's Windows share settings for the HP NAS 8000.

- outParam[0] = Comment

setNetworkSmbComment

Sets the comment associated with the network's Windows share settings for the HP NAS 8000.

- inParam[0] = Comment

getNetworkSmbWinsServerAddress

Similar to DNS, the Windows Internet Naming Service (WINS) is the Windows NT server method for associating a computer's host name with its address.

Returns the WINS server address.

- outParam[0] = WINS server IP address

setNetworkSmbWinsServerAddress

Sets the WINS server address.

- inParam[0] = WINS server IP address

getNetworkSmbSecurityMode

There are two Windows NT security modes:

Share-Level Security: The HP NAS 8000 handles its own security. Shares may be password-protected and may limit your access (read-only and/or read/ write) to data. You may define a password when you create the share.

User-Level Security: A domain controller is used to authenticate users when they access the HP NAS 8000. This requires specifying the domain name.

Returns the current SMB security mode.

- outParam[0] = SHARE or DOMAIN

setNetworkSmbSecurityMode

Sets the SMB Windows sharing security mode.

- inParam[0] = SHARE or DOMAIN
- inParam[1] = Domain/workgroup name

getNetworkSmbDomainControllerList

Returns the domain controller list for SMB Windows share settings.

- outParam[0:N] = Domain controller WINS names

setNetworkSmbDomainControllerList

If your SMB is controlled by specifying domain controllers, enter the NetBIOS name of the domain controller that is used to authenticate users. Do not include domain information with the NetBIOS name. You can specify more than one domain controller (backup domain controllers), but list them in the order you want them accessed.

Sets the SMB domain controller list for Windows sharing.

- inParam[0:N] = Domain controller WINS names

getNetworkSmbDomainList

Get the Windows domains that the NAS 8000 has been joined to.

- outParam[0:N] = Joined domain name list

getNetworkSmbDomainName

Get the name of the Windows domain that the NAS 8000 is currently joined to.

- outParam[0] = Domain/workgroup name

joinNetworkSmbDomain

Join the NAS 8000 to a specific Windows domain.

- inParam[0] = Domain name

removeNetworkSmbDomain

Removes the HP NAS 8000 from the specified SMB domain.

- inParam[0] = Domain name

getNetworkSmbDomainStatus

This call is only applicable when in user (domain) SMB security mode. The result is true only if the NAS 8000 is still a valid member of the domain. Technically this makes sure the shared secret between the NAS 8000 and domain controller is valid.

- outParam[0] = T or F

NFS Settings

Network File System (NFS) settings are optional. NFS is a client/server application that lets a UNIX computer user view and optionally store and update files on a remote computer as though they were on the user's own computer.

getStorageShareNfsEnabled

Identifies whether UNIX NFS sharing is enabled for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0] = T (Enabled) or F (Disabled)

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = T (Enabled) or F (Disabled)

setStorageShareNfsEnabled

Enables or disables UNIX NFS sharing for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- inParam[2] = T (Enable) or F (Disable)
- inParam[3] = NFS name (Only if enabling)

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- inParam[1] = T (Enable) or F (Disable)
- inParam[2] = NFS name (Only if enabling)

getStorageShareNfsName

Returns the UNIX NFS share name for the specified share.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0] = NFS name

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = NFS name

getStorageShareNfsReadOnly

Informs whether the specified share is set for UNIX NFS read-only or read/ write access.

- inParam[0] = Volume name
- inParam[1] = Share path
- outParam[0] = T (ReadOnly) or F (ReadWrite)

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- outParam[0] = T (ReadOnly) or F (ReadWrite)

setStorageShareNfsReadOnly

Sets the specified share to UNIX NFS read-only or read/write access.

- inParam[0] = Volume name
- inParam[1] = Share path
- inParam[2] = T (ReadOnly) or F (ReadWrite)

OR (see **getStorageShareList**). This facilitates webFilter usage.

- inParam[0] = Volume name:_*_:Share path
- inParam[1] = T (ReadOnly) or F (ReadWrite)

getNetworkNfsTrustedHostList

Displays the list of UNIX hosts allowed access to the system.

- outParam[0:N] = Trusted host list for NFS

setNetworkNfsTrustedHostList

Sets the IP address and host name of any trusted hosts. Root users on trusted hosts are given root user privileges on the HP NAS 8000.

- inParam[0:N] = Trusted host list for NFS

getNetworkNfsMaxConnections

Returns the maximum allowed number of simultaneous Network File System Daemons (NFSD) processes. Maximum allowed value is 128.

- outParam[0] = Maximum simultaneous NFS connections allowed

setNetworkNfsMaxConnections

Sets the maximum number of simultaneous Network File System Daemons (NFSD) processes. This controls the number of simultaneous NFS requests processed. The more NFSD processes you have, the more resources are used.

- inParam[0] = Maximum simultaneous NFS connections allowed

Admin Settings

The NAS server can be password protected. This prevents unauthorized access to the Command View NAS web interface. The NAS server ships without password protection, and the password must be set using the text or web interface. Additional admin settings are described in this section.

importSystemPasswdFile

Imports the specified file of system passwords.

- inParam[0] = Path name

importSystemGroupFile

Imports the specified file of system groups.

- inParam[0] = Path name

setSystemReservedDomainRange

Sets the range of UID and GID numbers that are used by the system for automatic user creation.

When a user connects to the system for the first time and is not a recognized user they are given a UID/GID from this range of numbers. This range should be outside of the range of numbers normally issued to users on the network.

- inParam[0] = Min
- inParam[1] = Max

getSystemReservedDomainRange

Returns the range of UID and GID numbers that are used by the system for automatic user creation.

- outParam[0] = Min - max

getSystemAdminDebugLevel

The admin debug level is a number which dictates the level of output the NAS 8000 puts in the system log.

- outParam[0] = Admin debug level

Options include:

- 0 = Standard amount of information is placed in the system logs
- 10 = Detailed information. Generally used for troubleshooting by HP support

setSystemAdminDebugLevel

- inParam[0] = Admin debug level

getSystemAdminPassword

Returns the system admin password.

- outParam[0] = Admin password

setSystemAdminPassword

Sets the system admin password.

- inParam[0] = Admin password

Local Admin Settings

These informational settings allow you to specify contact information and information about the physical location of your NAS 8000 solution.

getSystemLocalAdminAssetNumber

Returns the asset number assigned to the HP NAS 8000.

- outParam[0] = Asset number

setSystemLocalAdminAssetNumber

Defines the asset number assigned to the HP NAS 8000.

- inParam[0] = Asset number

getSystemLocalAdminContactName

Lists the name of the primary IT contact responsible for the HP NAS 8000.

- outParam[0] = Contact name

setSystemLocalAdminContactName

Sets the name of the primary IT contact responsible for the HP NAS 8000.

- inParam[0] = Contact name

getSystemLocalAdminContactPhone

Lists the phone number of the primary IT contact responsible for the HP NAS 8000.

- outParam[0] = Contact phone

setSystemLocalAdminContactPhone

Defines the phone number of the primary IT contact responsible for the HP NAS 8000.

- inParam[0] = Contact phone

getSystemLocalAdminContactEmail

Lists the email address of the primary IT contact responsible for the HP NAS 8000.

- outParam[0] = Contact email

setSystemLocalAdminContactEmail

Defines the email address of the primary IT contact responsible for the HP NAS 8000.

- inParam[0] = Contact email

getSystemLocalAdminContactPager

Returns the pager number of the primary IT contact responsible for the HP NAS 8000.

- outParam[0] = Contact pager

setSystemLocalAdminContactPager

Defines the pager number of the primary IT contact responsible for the HP NAS 8000.

- inParam[0] = Contact pager

getSystemLocalAdminLocation

Returns a description of the physical location of the HP NAS 8000 system.

- outParam[0] = Location

setSystemLocalAdminLocation

Defines the description of the physical location of the HP NAS 8000 system.

- inParam[0] = Location

getSystemLocalAdminRackId

Returns the rack ID for the HP NAS 8000 system.

- outParam[0] = Rack ID

setSystemLocalAdminRackId

Defines the rack ID for the HP NAS 8000 system.

- inParam[0] = Rack ID

getSystemLocalAdminRackPosition

Returns the rack position for the HP NAS 8000 system.

- outParam[0] = Rack position

setSystemLocalAdminRackPosition

Defines the rack position for the HP NAS 8000 system.

- inParam[0] = Rack position

System Settings

The commands in this section are used to defined general system descriptions and settings for the NAS 8000.

System Description

getSystemOid

Returns the OID for the NAS device (.1.3.6.1.4.1.11.10.2.1.9.1)

- outParam[0] = .1.3.6.1.4.1.11.10.2.1.9.1

getSystemGlobalUniqueId

Returns the Global Unique ID for this device (a string of characters that uniquely identifies this unit)

- outParam[0] = Global unique ID - per SEMI 1.0

getSystemManagementUrl

Returns the Management URL for this SEMI-compatible device.

- outParam[0] = Management URL

getNetworkSnmpActiveTrapDestList

Returns the complete list of “active” Trap Destinations.

- outParam[0:n] = Management URL

System Events

getSystemHealth

Returns the current overall Health status for the system.

- outParam[0] = Health status (numeric)
- outParam[1] = Health status (string)

Status messages include:

- OK
- Warning
- Critical

setSystemEvent

Creates a System Event. Should be the proper size, no filler added.

- inParam[0] = Event (hpnSystemEvent)
- inParam[1:2k] = vblVal (stringsMap) (pair of strings)

clearSystemEvent

Clears a System Event. The Master Event Daemon (hpMed) will then handle things properly, sending e-mail as well as the SNMP traps to the various targets.

getSystemEventNum

Get the event number for a given event name.

- inParam[0] = Event name
- outParam[0] = Event number

getSystemEventName

Get the name of a given event number.

- inParam[0] = Event number
- outParam[0] = Event name CPU settings

getSystemCpuList

Returns a list of the CPUs installed in the NAS server.

- outParam[0:N] = CPU number

getSystemCpuCurrentLoadPercent

Returns information about the current load on the specified CPU.

- inParam[0] = CPU number
- outParam[0] = Current load percent

getSystemCpuPeakLoadPercent

Returns information about the peak load on the specified CPU.

- inParam[0] = CPU number
- outParam[0] = Peak load percent

getSystemCpuAverageLoadPercent

Returns information about the average load on the specified CPU.

- inParam[0] = CPU number
- outParam[0] = Average load percent

Time and Date Settings

The HP NAS 8000 uses the information on this screen to keep track of the date and time for operations such as time stamps for file generation and modification. Failure to set the proper date and time may lead to confusing behavior or misleading time stamping of files and log messages.

getSystemTimestamp

Returns the system timestamp.

- inParam[0] = Optional format string
- outParam[0] = Date & Time according to format

Common formats follow. Default format is "%m/%d/%Y %T".

- %c - TIME & DATE for locale
- %x - DATE for locale
- %X - TIME for locale
- %r - Time am/pm (exact format may change with locale)
- %R - %H:%M
- %T - %H:%M:%S (24 hour)
- %D - %m/%d/%y - (United States specific format)

setSystemTimestamp

Sets the system timestamp.

- inParam[0] = Date (mm/dd/yyyy) or "NOW"
- inParam[1] = Time (24hh:mm:ss) or "NOW"

getSystemDate

Returns the current system date.

- inParam[0] = Optional format string
- outParam[0] = Date according to format (default is "%m/%d/%Y")

Common formats follow. Default format is "%m/%d/%Y %T".

- %c - TIME & DATE for locale
- %x - DATE for locale
- %X - TIME for locale
- %r - Time am/pm (exact format may change with locale)
- %R - %H:%M
- %T - %H:%M:%S (24 hour)
- %D - %m/%d/%y - (United States specific format)

setSystemDate

Sets the system date.

- inParam[0] = Date [MM/DD/YYYY] or “NOW”

getSystemTime

Displays the current system time.

- inParam[0] = optional format string
- outParam[0] = Time according to format (default is “%T”)

Common formats follow. Default format is “%m/%d/%Y %T”.

- %c - TIME & DATE for locale
- %x - DATE for locale
- %X - TIME for locale
- %r - Time am/pm (exact format may change with locale)
- %R - %H:%M
- %T - %H:%M:%S (24 hour)
- %D - %m/%d/%y - (United States specific format)

setSystemTime

Sets the current system time.

- inParam[0] = Time [24HH:MM:SS] or “NOW”

getSystemTimezoneCurrentGmt

- outParam[0] = GMT value

getSystemTimezoneEnabled

Informs whether time zones are enabled on the HP NAS 8000.

- inParam[0] = Time zone name
- outParam[0] = Enable status

setSystemTimezoneEnabled

Enables time zone support.

- inParam[0] = Time zone name
- inParam[1] = Enable

getSystemTimezoneCurrentDescription

Displays the current time zone setting description.

- inParam[0] = Time zone name
- outParam[0] = Time zone description

getSystemTimezoneCurrentCountry

Displays the current time zone country.

- inParam[0] = Time zone name
- outParam[0] = Time zone country

getSystemTimezoneList

Displays the available time zone names.

- outParam[0:N] = Time zone names

getSystemTimezoneCurrentName

Displays the current system time zone.

- inParam[0] = Time zone name

getSystemTimezoneCurrentGmt

Displays the offset from GMT for the current system time zone.

- outParam[0] = Offset in hours

Example: -0700

System NTP Settings

The following commands are used to set the Network Time Protocol (NTP) and choose a server with which the NAS 8000 can synchronize system time.

addSystemNtpServer

Adds a server to the list of hosts from which to get NTP time synchronization information.

- InParam[0] = Hostname or NTP server IP address

getSystemNtpEnabled

Determine whether the server is configured to sync its system time with NTP servers.

- outParam[0] = T or F

getSystemNtpServerList

Gets the list of NTP servers from which the system is currently getting NTP time synchronization information.

- outParam[0:N] = List of NTP server hostnames/IP addresses

setSystemNtpEnabled

Enables or disables NTP time synchronization

- inParam[0] = T or F

Email Settings

The HP NAS 8000 lets you automatically notify individuals via email if there is a hardware failure or a critical system alert.

Note Email settings are optional. In the event of a hardware failure or system alert, messages are sent through the network management tool or they are logged in the system log.

getSystemEmailRecipientAddressList

Returns a list of the email addresses for the recipients of any system alert messages that may be sent.

- outParam[0:N] = Email recipient address list

setSystemEmailRecipientAddressList

Sets the email addresses for the recipients of any system alert messages that may be sent.

- inParam[0:N] = Email recipient address list

getSystemEmailServerName

Returns the name of the SMTP mail server.

- outParam[0] = Email server name

setSystemEmailServerName

Sets the SMTP mail server. The email server must be an IP address or a fully qualified name (such as alpha.corp.com)

- inParam[0] = Email server name

sendSystemEmail

Sends the specified email message to the email recipient list.

- inParam[0] = Email address
- inParam[1] = Email subject
- inParam[2] = Email body

sendSystemEmailTest

Sends a test email to the specified email address.

- inParam[0] = Email address

getSystemNameEmailAddress

Returns the email address of the specified user.

- inParam[0] = User name
- outParam[0] = Email address

getSystemUIDEmailAddress

Returns the email address for the user with the specified ID.

- inParam[0] = UID
- outParam[0] = Email address

setSystemNameEmailAddress

Sets the email address for the specified user.

- inParam[0] = User name
- inParam[1] = Email address

setSystemUidEmailAddress

Sets the email address for the user with the specified ID.

- inParam[0] = UID
- inParam[1] = Email address

Log Settings

You can redirect a copy of the system log to a server you choose. This allows you to manage a central location for the event log instead of working with different interfaces or systems.

Note Check your syslog server's documentation for instructions on enabling syslog event reception from network clients.

getSystemLogEntryDescription

Returns a simple list of the entries that exist in the system log.

- outParam[0:N] = Log file entry handles

getSystemLogEntryList

Returns a detailed list of the entries that exist in the system log.

- inParam[0] = Log entry handle
- inParam[1] = String to quote (optional)
- inParam[2] = String used to quote (optional)
- outParam[0] = Log entry description

getSystemLogEntryProgramName

Returns the name of the program that created the log entry.

- inParam[0] = Log entry handle
- outParam[0] = Log entry program name

getSystemLogEntrySeverity

Returns the severity level associated with the specified log entry.

- inParam[0] = Log entry handle
- outParam[0] = Log entry severity

Values:

- E = Error
- W = Warn
- I = Info
- U = Unknown

getSystemLogEntryTimestamp

Returns the time that the event triggering the specified log entry occurred.

- inParam[0] = Log entry handle
- inParam[1] = Optional format string
- outParam[0] = Log entry date/time

Default format is "%m/%d/%Y %T".

- %c - TIME & DATE for locale
- %x - DATE for locale
- %X - TIME for locale
- %r - Time am/pm (exact format may change with locale)
- %R - %H:%M
- %T - %H:%M:%S (24 hour)
- %D - %m/%d/%y - (United States specific format)
- outParam[0] = Time [HH:MM:SS]

appendSystemLogEntry

Adds the specified entry to the system log.

- inParam[0] = Severity
- inParam[1:N] = Log message

Valid severity values:

- 0 = Error, E
- 1 = Warning, W
- 2 = Info, I

getSystemSyslogServerName

Returns the name of the remote syslog server.

- outParam[0] = Syslog server name

setSystemSyslogServerName

Defines the remote server to which system log entries will be sent.

- inParam[0] = Syslog server name

UPS Settings

If you connected the optional APC Symmetra UPS to your HP NAS 8000 during installation, the NAS server attempts to communicate with the UPS through a serial connection. The APC UPS has only one serial port. The NAS server, however, has two ports. Connect the UPS serial port to the COM1 port on the back of the NAS server.

getSystemUpsStatus

Returns the status of the attached supported UPS.

- outParam[0] = Status
- outParam[0] = Status

Values:

- normal: UPS support is currently disabled.
- APC UPS is normal: UPS support is enabled for APC and is functioning normally.
- APC UPS fail to attach: UPS support is enabled for APC but the system cannot communicate with the UPS.
- APC UPS comm is pending: UPS support is enabled for APC. The system is in the process of establishing communications.
- APC UPS unknown error:xxxx: An unexpected error has occurred. “xxxx” will display information about the error.

getSystemUpsList

Returns a list of the names of supported UPS systems.

- outParam[0:N] = List of UPS names

getSystemUpsEnabled

Returns the name of the attached enabled UPS.

- outParam[0] = Name of enabled UPS

setSystemUpsEnabled

Sets the name of the attached enabled UPS.

- inParam[0] = Name of UPS

getSystemUpsTitle

Returns the system title of the specified UPS.

- inParam[0] = Name of UPS
- outParam[0] = Title of UPS

General System Settings

These commands are used to review general system information and to reboot or shut down the NAS server.

getSystemManufacturer

Returns the system manufacturer name (Hewlett-Packard).

- outParam[0] = Manufacturer

getSystemOsBuildNumber

Returns the build number of the operating system in the active boot partition.

- OutParam[0] = Build number

getSystemOsVersion

Returns the version number for the OS installed on the NAS server.

- outParam[0] = OS version

getSystemProductAssetTag

Returns the asset tag assigned to the HP NAS 8000 system.

- outParam[0] = Asset tag

getSystemProductName

Returns the product name.

- outParam[0] = Product name

getSystemProductNumber

Returns the product number.

- outParam[0] = Product number

getSystemNasSerialNumber

Returns the HP NAS 8000 solution's serial number.

- outParam[0] = Serial number

getSystemMotherboardSerialNumber

Returns the serial number of the NAS server mother board.

- OutParam[0] = Serial number

getSystemShortProductName

Returns the short version of the product name.

- outParam[0] = Short product name

getSystemUptime

Returns the amount of time the system has been functioning since the last power up.

- outParam[0] = System uptime

doSystemReboot

Reboots the NAS server.

doSystemShutdown

Shuts down the NAS server.

Software Module Settings

You can enable/disable the following software modules for use with the HP NAS 8000:

- Data snapshots
- Virus protection
- Tape backup

Before use, software modules must be installed, authorized and enabled. The following commands install, uninstall, enable and disable these software modules.

getSystemPackageAuthorized

Determines whether the specified add-on software modules is installed and unlocked on the NAS server.

- inParam[0] = Package name
- outParam[0] = Package enable

getSystemPackageDescription

Returns the description of the specified add-on software module.

- inParam[0] = Package name
- outParam[0] = Package description

getSystemPackageEnabled

Informs whether the specified add-on software module.

- inParam[0] = Package name
- outParam[0] = Package enable

setSystemPackageEnabled

Enables or disables the specified add-on software module.

- inParam[0] = Package name
- inParam[1] = Enable

getSystemPackageList

Lists the add-on software modules available on the NAS server.

- [0:N] = Package names

getSystemPackageName

Returns the name of the specified add-on software module.

- inParam[0] = Package name
- outParam[0] = Package name

getSystemPackageVersion

Returns the version of the specified add-on software module.

- inParam[0] = Package name
- outParam[0] = Package version

getSystemPackageWebpage

Returns the URL for accessing the web page with information about the specified add-on software module.

- inParam[0] = Package name
- outParam[0] = Package Web page

setSystemPackageWebpage

Sets the URL for accessing the web page with information about the specified add-on software module.

- inParam[0] = Package name
- inParam[1] = Web page

installSystemPackage

Installs the specified add-on software module from the specified URL.

- inParam[0] = Package URL
- inParam[1] = Proxy name (optional)
- inParam[2] = Proxy number (optional)

uninstallSystemPackage

Uninstalls the specified add-on software module.

- inParam[0] = Package name

User and Group Settings

The following commands allow you to map users of the Server Message Block Protocol/ Common Internet File System protocol (SMB/CIFS - Windows) to users of the Network File System protocol (NFS - UNIX).

The HP NAS 8000 maintains a mapping of users between the two protocols. If a Windows user is not mapped to an existing UNIX user ID, a new UNIX user ID is generated and the Windows user is mapped to it when the Windows user accesses the HP NAS 830 for the first time.

Mapping users improves adherence to file and directory permissions and compliance to disk quotas.

In addition to setting up user mapping, you can export a list of Windows and UNIX users. You can also import or export a user map file. Importing a user map lets an unlimited number of mappings occur simultaneously. Exporting a user map lets you save the map for later use or for disaster recovery.

Note User mapping is available only when you select user-level security on the Windows Security screen. User mapping using names is most useful when NIS is enabled.

getSystemUnixUserNamesList

Returns a list of UNIX users on the system.

- outParam[0:1] = UNIX user name

getSystemUnixGroupNamesList

Returns a list of UNIX groups on the system.

- outParam[0:N] = UNIX group name

getSystemUserMapList

Returns a list of mapped users, with both Windows and UNIX user IDs listed.

- outParam[0:N] = Windows user -> UNIX user

getSystemUserMapList

Returns a list of mapped users, with both Windows and UNIX user IDs listed.

- outParam[0:N] = Windows user -> UNIX user

getSystemGroupMapList

Returns a list of mapped groups, with both Windows and UNIX group names listed.

- outParam[0:N] = Windows group -> UNIX group

getSystemWindowsDomainsList

Returns a list of the current and trusted Windows domains.

- outParam[0:N] = Windows domains

getSystemWindowsUserNamesFromDomainList

Returns a list of users' Windows IDs for users on the specified Windows domain.

- inParam[0] = Domain
- outParam[0:N] = Windows user

getSystemWindowsGroupNamesFromDomainList

Returns a list of mapped users, with both Windows and UNIX user IDs listed, for the specified Windows domain.

- inParam[0] = Domain
- outParam[0:N] = Windows user -> UNIX user

mapSystemWindowsUserNameToUnixUserName

Maps the specified Windows user name to the specified UNIX user name and ID.

- inParam[0] = Windows user name
- inParam[1] = unix name | unix name:uid | uid

mapSystemWindowsGroupNameToUnixGroupName

Maps the specified Windows group name to the specified UNIX group name and ID.

- inParam[0] = Windows group name
- inParam[1] = unix group | unix group:gid | gid

unmapSystemWindowsUserNameFromUnixUserName

Unmaps the specified Windows user name from the specified UNIX domain and user name.

- inParam[0] = Windows user name
- inParam[1] = UNIX name DOMAIN+unixuser

unmapSystemWindowsGroupNameFromUnixGroupName

Unmaps the specified Windows group name from the specified UNIX domain and user name.

- inParam[0] = Windows group name
- inParam[1] = UNIX name DOMAIN+unixuser

User and Group Quota Settings

Quotas allow you to restrict the space usage on the HP NAS 8000 for both users and groups. You also can enable default quotas for users and/or groups. The default quota applies to all users or groups who do not have a specific quota assigned to them. A user or group goes beyond their defined space usage cannot write additional data to the system. Each user can have one quota on a file volume.

Quotas are set on a per file volume basis. You must enable a file volume for quotas before the settings can be modified. You can have a quota larger than the available space on the file volume.

All user list files and quota files are in text form to allow for easy editing and script conditioning.

getSystemGroupQuota

Returns information about the quota defined for the specified group on the specified file volume.

- inParam[0] = Group name
- inParam[1] = Volume name
- outParam[0] = Current usage Mb
- outParam[1] = Soft limit
- outParam[2] = Hard limit
- outParam[3] = Grace period

setSystemGroupQuota

Sets the quota settings for the specified group on the specified file volume.

- inParam[0] = Group name
- inParam[1] = Volume name
- inParam[2] = Soft limit
- inParam[3] = Hard limit

getSystemUserQuota

Returns information about the quota defined for the specified user on the specified file volume.

- inParam[0] = User name
- inParam[1] = Volume name
- outParam[0] = Current usage Mb
- outParam[1] = Soft limit
- outParam[2] = Hard limit
- outParam[3] = Grace period

setSystemUserQuota

Sets the quota settings for the specified user on the specified file volume.

- inParam[0] = User name
- inParam[1] = Volume name
- inParam[2] = Soft limit
- inParam[3] = Hard limit

getSystemUserQuotaList

Returns a list of user quotas for the specified file volume.

- inParam[0] = Volume name
- outParam[0:N] = Quota

getSystemGroupQuotaList

Returns a list of group quotas for the specified file volume.

- inParam[0] = Volume name
- outParam[0:N] = Quotas

getSystemUserGracePeriod

Displays the grace period – the amount of time users will have access to the space after their quota is reached – for the specified file volume. The time a user who has exceeded their soft quota has to remove data and bring their storage usage under their soft quota. If the grace period runs out, the user will be blocked from any further writes.

- inParam[0] = Volume name
- outParam[0] = Grace period in seconds

setSystemUserGracePeriod

Sets the grace period – that is the amount of time users will have access to the space after their quota is reached – for the specified file volume.

- inParam[0] = Volume name
- inParam[1] = Grace period in seconds

getSystemGroupGracePeriod

Displays the grace period – that is the amount of time groups will have access to the space after their quota is reached – for the specified file volume.

- inParam[0] = Volume name
- outParam[0] = Grace period in seconds

setSystemGroupGracePeriod

Sets the grace period – that is the amount of time groups will have access to the space after their quota is reached – for the specified file volume.

- inParam[0] = Volume name
- inParam[1] = Grace period in seconds

High Availability Cluster Settings

The following commands let you create and control a high-availability cluster.

setSystemHostsFileEntry

Adds an entry to the system hosts file list.

- inParam[0] = Host name
- inParam[1] = IP address
- inParam[2] = Domain name

deleteSystemHostsFileEntry

Removes the designated entry from the system hosts file list.

- inParam[0] = Host name

applyClusterConfiguration

Creates a cluster using either default values or values set by other **setCluster*** commands. After this completes, the nodes are considered to be in a cluster.

When this command is executed, the cluster is defined, but it is not started and no packages are running. Registries on all the nodes are synchronized using the node on which the command was executed as the master.

deleteClusterConfiguration

Deletes the cluster configuration, including package configurations and restores the registry to default values. Nodes are now running as separate NAS servers.

startCluster

Starts a cluster after it has been created with the **applyClusterConfiguration** command.

Starting a cluster causes three things to happen:

- Synchronizes the registries of all nodes using the node on which the **startCluster** command was executed, as the master.
- Deactivates storage on all the nodes. This guarantees that volume groups, volumes and shares are active on only one node and under control of a package.
- Starts all the packages that have been configured.

stopCluster

Stops the cluster and all packages. The nodes are still considered to be in a cluster, but the cluster is down.

startClusterNode

Brings a node that was previously stopped back into the cluster.

- InParam[0] = Node name

stopClusterNode

Takes a node out of the cluster and transfers all it's running packages to an alternate node.

- InParam[0] = Node name

getClusterName

Gets the name of the cluster stored in the registry.

- OutParam[0] = Cluster name

setClusterName

Stores the name in the registry. Does not take effect until **applyClusterConfiguration** is called.

- InParam[0] = Cluster name

getClusterStatus

Gets status information about the cluster, nodes, and packages.

- OutParam[0] = A multi-line text string that contains status information about the cluster, including nodes and packages running on those nodes.

getClusterQsHost

Gets the name of the Quorum server stored in the registry.

- OutParam[0] = Quorum server name

setClusterQsHost

Stores the name of the Quorum server in the registry. This command is required before executing **applyClusterConfiguration**.

- InParam[0] = Cluster name

getClusterAutoStartEnabled

Gets the **AutoStartEnabled** value from the registry,

- OutParam[0] = T (enabled), or F (disabled)

setClusterAutoStartEnabled

Stores the value in the registry and also updates the cluster configuration to reflect the change. If enabled, each node will attempt to join the cluster at boot time. If disabled, the cluster must be started manually using **startCluster**.

- InParam[0] = T (enabled), or F (disabled)

getClusterNodeNameList

Gets the list of nodes in the cluster.

- OutParam[0:N] = List of node names

setClusterNodeNameList

Stores the list of nodes in the cluster in the registry.

- InParam[0:N] = List of node names

getClusterHeartbeatNetworkCardList

Gets the list of network card interfaces used for heartbeats as they are stored in the registry.

- OutParam[0:N] = Network card interface list

setClusterHeartbeatNetworkCardList

Stores the list in the registry. Is required before calling **applyClusterConfiguration**.

- InParam[0:N] = List of heartbeat network cards

For example:

- setClusterHeartbeatNetworkCardList eth0 eth1

getClusterQsPollingInterval

Gets the interval from the registry.

- OutParam[0] = Interval, in seconds, for contacting the Quorum server

setClusterQsPollingInterval

Stores the polling interval in the registry: how often, in seconds, to contact Quorum server.

- InParam[0] = Interval, in seconds

getClusterHeartbeatInterval

Gets the interval from the registry.

- OutParam[0] = Interval, in seconds, for sending heartbeats

setClusterHeartbeatInterval

Stores the interval in the registry.

- InParam[0] = Interval, in seconds, for sending heartbeats

getClusterNodeTimeout

Gets the timeout from the registry.

- OutParam[0] = Time period, in seconds, before a node is considered gone from the cluster

setClusterNodeTimeout

Stores the timeout in the registry: how long to wait, in seconds, before considering a node gone from the cluster.

- InParam[0] = Interval

getClusterAutoStartTimeout

Gets the timeout from the registry.

- OutParam[0] = Time period, in seconds, before timeout occurs

setClusterAutoStartTimeout

Stores the timeout in the registry. (How long to wait, in seconds, before timeout occurs.)

- InParam[0] = Interval, in seconds

getClusterNetworkPollingInterval

Gets the timeout from the registry.

- OutParam[0] = Time period, in seconds, before polling occurs

setClusterNetworkPollingInterval

Stores the timeout in the registry: how long to wait, in seconds, before polling.

- InParam[0] = Interval, in seconds, between polling events

getClusterMaxConfiguredPackages

Gets the value from the registry.

- OutParam[0] = Maximum number of packages allowed to run

setClusterMaxConfiguredPackages

Stores the number in the registry.

- InParam[0] = Maximum number of packages allowed to run

getClusterNtpLocalMaster

Stores the number in the registry.

- OutParam[0] = Hostname or IP address of cluster node that is currently serving as an NTP master server for time synchronization.

setClusterNtpLocalMaster

Assigns one of the cluster nodes to be an NTP master server for time synchronization. Establishes a stratum 10 NTP server on one node, other nodes are made to synchronize with the local master.

- InParam[0] = Cluster node name
- InParam[1] = Cluster node IP address

getClusterCallTimeout

Returns the value from the registry.

- OutParam[0] = Timeout, in seconds

restartClusterAntivirus

Executes **restartClusterNodeAntivirus** on all cluster nodes.

restartClusterNodeAntivirus

Restarts the antivirus software.

Cluster Package Commands

The following commands let you configure and control failover packages.

addClusterPackage

Creates a default package configuration and a placeholder in the registry. This must be called before other calls to setClusterPackageXYZ text commands. The package is defined in the registry, with some default values, but not defined in the cluster.

- InParam[0] = Package name

applyClusterPackageConfiguration

Creates a cluster package using either default values or values set by other setClusterPackageXYZ commands. After this completes, the package is known to the cluster, but not running.

- InParam[0] = Package name

startClusterPackage

Starts the package on the primary owning node, or if *nodeName* is given, then on that node. The volume group(s) associated with the package are activated on the node running the package. If the volume groups have previously defined volumes and shares, then those are activated.

- InParam[0] = Package name
- InParam[1] = Node name

stopClusterPackage

Stops the package wherever it is running. The volume group(s) and all associated storage that belong to the package are deactivated on the node running the package.

- InParam[0] = Package name

deleteClusterPackage

Deletes the package configuration from the cluster and the registry. The volume group(s) and all associated storage that belonged to the package are not affected. They are deactivated, but still exist.

- InParam[0] = Package name

relocateClusterPackage

Stops the package on the current node and starts it up on the given node. Sets the value of PrimaryOwner to the new given node and applies the updated configuration.

- InParam[0] = Package name
- InParam[1] = Node name

failoverClusterPackage

Stops the package on the current node and starts it up on the given node.

- InParam[0] = Package name
- InParam[1] = Node name

failbackClusterPackag

Stops the package on the current node and starts it up on the Primary owning node.

- InParam[0] = Package name

setClusterPackageEnabled

Enables or disables the package from running on the given node. If a package is disabled from running on an alternate node, then it cannot failover to that node.

- InParam[0] = Package name
- InParam[1] = Node name
- InParam[2] = T (enabled) or F (disabled)

getClusterPackageNameList

Gets the complete list of package names that have been created on the cluster, whether running or not.

- OutParam[0:N] = List of package names

getClusterPackagePrimaryOwner

Gets from the registry, the name of the node that owns the package.

- InParam[0] = Package name
- OutParam[0] = Primary owning node of the package

setClusterPackagePrimaryOwner

Required. Sets in the registry, the name of the node that owns the package. Running '**startClusterPackage** packageName' without a nodeName argument, starts the package on the primary owning node.

- InParam[0] = Package name
- InParam[1] = Node name

getClusterPackageCurrentOwner

Gets the name of the node that is currently running the package.

- InParam[0] = Package name
- OutParam[0] = Node name

getClusterPackageAdoptiveNodeList

Gets a list of the nodes that are allowed to run the package in a failover situation.

- InParam[0] = Package name
- OutParam[0:N] = Node name list

getClusterPackageVirtualIpAddressList

Gets from the registry, the VirtualIpAddress/Subnet values served by this package.

- InParam[0] = Package name
- OutParam[0:N] = VirtualIPAddress/Subnet list

setClusterPackageVirtualIpAddressList

Required. Sets the given VirtualIpAddress/Subnet list for the given package.

- InParam[0] = Package name
- InParam[1:N] = VirtualIpAddress/Subnet entries

For example:

- setClusterPackageVirtualIpAddressList pkg1 192.168.100.21/192.168.100.0

getClusterPackageVolumegroupList

Gets from the registry, the list of volume groups managed by this package.

- InParam[0] = Package name
- OutParam[0:N] = Volume group list

setClusterPackageVolumegroupList

Required. Sets in the registry, the list of volume groups to be managed for the given package.

- InParam[0] = Package name
- InParam[1:N] = Volume group list

For example:

- setClusterPackageVolumegroupList pkg1 vg1 vg2 vg3

getClusterPackageFailoverPolicy

Gets from the registry, value of the failover policy for this package.

- InParam[0] = Package name
- OutParam[0] = CONFIGURED_NODE or MIN_PACKAGE_NODE

setClusterPackageFailoverPolicy

Sets in the registry, the failover policy for the given package. The default is CONFIGURED_NODE.

- InParam[0] = Package name
- InParam[1] = CONFIGURED_NODE or MIN_PACKAGE_NODE

getClusterPackageFailbackPolicy

Gets from the registry, value of the failback policy for this package.

- InParam[0] = Package name
- OutParam[0] = MANUAL or AUTOMATIC

setClusterPackageFailbackPolicy

Sets in the registry, the failover policy for the given package. The default is MANUAL.

- InParam[0] = Package name
- InParam[1] = MANUAL or AUTOMATIC

getClusterPackageAutoRun

Gets from the registry, value of AutoRun for this package.

- InParam[0] = Package name
- OutParam[0] = T or F

setClusterPackageAutoRun

Sets in the registry, the AutoRun value for the given package. The default is T.

- InParam[0] = Package name
- InParam[1] = T or F

getClusterPackageNodeFailFastEnabled

Gets from the registry, value of NodeFailFastEnabled for this package.

- InParam[0] = Package name
- OutParam[0] = T or F

setClusterPackageNodeFailFastEnabled

Sets in the registry, the NodeFailFastEnabled value for the given package. The default is F.

- InParam[0] = Package name
- InParam[1] = T or F
- OutParam[0] = None

getClusterPackageServiceFailFastEnabled

Gets from the registry, the value of ServiceFailFastEnabled for this package.

- InParam[0] = Package name
- OutParam[0] = T or F

setClusterPackageServiceFailFastEnabled

Sets in the registry, the ServiceFailFastEnabled value for the given package. The default is F.

- InParam[0] = Package name
- InParam[1] = T or F
- OutParam[0] = None

getClusterPackageServiceHaltTimeout

Gets from the registry, the value of service halt timeout for this package.

- InParam[0] = Package name
- OutParam[0] = Number, in seconds

setClusterPackageServiceHaltTimeout

Sets in the registry, the ServiceHaltTimeout value for the given package. The default is 30 seconds.

- InParam[0] = Package name
- InParam[1] = Time, in seconds

Cluster Operation

The following text commands are available for cluster operations. These operations are largely a simple "wrapping" of the basic text command, but allow administration to be done from one machine and the operations mirrored on other nodes in the cluster.

setClusterSystemReservedDomainRange

Executes **setSystemReservedDomainRange** on all cluster nodes.

- InParam[0] = Min
- InParam[1] = Max

mapClusterSystemWindowsUserNameToUnixUserName

Executes **mapSystemWindowsUserNameToUnixUserName** on all cluster nodes.

- InParam[0] = Windows user name
- InParam[1] = unix name | unix name:uid | uid

mapClusterSystemWindowsGroupNameToUnixGroupName

Executes **mapClusterSystemWindowsGroupNameToUnixGroupName** on all cluster nodes.

- InParam[0] = Windows user name
- InParam[1] = unix group | unix group:gid | gid

unmapClusterSystemWindowsUserNameFromUnixUserName

Executes **unmapSystemWindowsUserNameFromUnixUserName** on all cluster nodes.

- InParam[0] = Windows user name
- InParam[1] = UNIX name DOMAIN+unixuser

unmapClusterSystemWindowsGroupNameFromUnixGroupName

Executes **unmapSystemWindowsGroupNameFromUnixGroupName** on all cluster nodes.

- InParam[0] = Windows user name
- InParam[1] = UNIX group DOMAIN+group
- OutParam[0] = None

setClusterNetworkSmbComment

Executes **setNetworkSmbComment** on all cluster nodes.

- inParam[0] = Comment

setClusterNetworkSmbWinsServerAddress

Executes **setNetworkSmbWinsServerAddress** on all cluster nodes. If no address is provided, the current WINS server address is cleared.

- InParam[0] = WINS server name | WINS server IP address

setClusterNetworkSmbSecurityMode

Executes **setNetworkSmbSecurityMode** on all cluster nodes.

- InParam[0] = SHARE or DOMAIN

setClusterNetworkSmbDomainControllerList

Executes **setNetworkSmbDomainControllerList** on all cluster nodes.

- inParam[0:N] = Domain controller WINS names

joinClusterNetworkSmbDomain

Executes **joinNetworkSmbDomain** on all cluster nodes.

- InParam[0] = Domain name

removeClusterNetworkSmbDomain

Executes **removeNetworkSmbDomain** on all cluster nodes.

- InParam[0] = Domain name

setClusterSystemNtpEnabled

Executes **setSystemNtpEnabled** on all cluster nodes.

- InParam[0] = T (enabled), or F (disabled)

addClusterSystemNtpServer

Executes **addSystemNtpServer** on all cluster nodes.

- InParam[0] = Hostname or NTP server IP address

removeClusterSystemNtpServer

Executes **removeSystemNtpServer** on all cluster nodes.

- InParam[0] = Server name or IP address

setClusterNetworkDnsDomainName

Executes **setNetworkDnsDomainName** on all cluster nodes.

- InParam[0] = Domain name

setClusterNetworkDnsAddressList

Executes **setNetworkDnsAddressList** on all cluster nodes.

- inParam[0:N] = DNS address list

setClusterNetworkNisDomainName

Executes **setNetworkNisDomainName** on all cluster nodes.

- InParam[0] = NIS domain name

setClusterNetworkNisEnabled

Executes setNetworkNisEnabled on all cluster nodes.

- InParam[0] = T (enabled), or F (disabled)

setClusterNetworkNisServerName

Executes **setNetworkNisServerName** on all cluster nodes.

- InParam[0] = NIS server name

setClusterNetworkNfsTrustedHostList

Executes **setNetworkNfsTrustedHostList** on all cluster nodes.

- inParam[0:N] = List of rusted host names or IP addresses for NFS
- OutParam[0] = None

setClusterNetworkNfsMaxConnections

Executes **setNetworkNfsMaxConnections** on all cluster nodes.

- InParam[0] = Max

setClusterSystemAdminDebugLevel

Executes **setSystemAdminDebugLevel** on all cluster nodes.

- InParam[0] = Debug level

setClusterSystemAdminPassword

Executes **setSystemAdminPassword** on all cluster nodes.

- InParam[0] = Password

setClusterSystemTimestamp

Executes **setSystemTimestamp** on all cluster nodes.

- InParam[0] = Date (mm/dd/yyyy or “NOW”)
- InParam[1] = Time (24hh:mm:ss or “NOW”)

setClusterSystemTimezoneEnabled

Executes **setSystemTimezoneEnabled** on all cluster nodes.

- InParam[0] = Time zone name
- InParam[1] = T (enabled), or F (disabled)

setClusterSystemUpsEnabled

Executes **setSystemUpsEnabled** on all cluster nodes.

- InParam[0] = UPS name

importClusterSystemPasswdFile

Executes **importSystemPasswdFile** on all cluster nodes. The import file on the execution machine is rsync'd to other cluster nodes and imported locally on all nodes.

- InParam[0] = Local pathname of import file

importClusterSystemGroupFile

Executes **importSystemGroupFile** on all cluster nodes. The import file on the execution machine is rsync'd to other cluster nodes and imported locally on all nodes.

- InParam[0] = Local pathname of import file

importClusterSystemUserMapList

Executes **importSystemUserMapList** on all cluster nodes. The import file on the execution machine is rsync'd to other cluster nodes and imported locally on all nodes.

- InParam[0] = Local pathname of import file

importClusterSystemGroupMapList

Executes **importSystemGroupMapList** on all cluster nodes. The import file on the execution machine is rsync'd to other cluster nodes and imported locally on all nodes.

- InParam[0] = Local pathname of import file

setClusterSystemEmailRecipientAddressList

Executes **setSystemEmailRecipientAddressList** on all cluster nodes.

- InParam[0:N] = List of email addresses

setClusterSystemEmailServerName

Executes **setSystemEmailServerName** on all cluster nodes.

- InParam[0] = email server

setClusterSystemLocalAdminAssetNumber

Executes **setSystemLocalAdminAssetNumber** on all cluster nodes.

- InParam[0] = Asset number

setClusterSystemLocalAdminContactName

Executes **setSystemLocalAdminContactName** on all cluster nodes.

- InParam[0] = Administrator contact name

setClusterSystemLocalAdminContactPhone

Executes **setSystemLocalAdminContactPhone** on all cluster nodes.

- InParam[0] = Administrator contact phone

setClusterSystemLocalAdminContactPager

Executes **setSystemLocalAdminContactPager** on all cluster nodes.

- InParam[0] = Administrator contact pager

setClusterSystemLocalAdminContactEmail

Executes **setSystemLocalAdminContactEmail** on all cluster nodes.

- InParam[0] = Administrator contact email

setClusterSystemLocalAdminLocation

Executes **setClusterSystemLocalAdminLocation** on all cluster nodes.

- InParam[0] = Administrator contact location

setClusterSystemLocalAdminRackId

Executes **setSystemLocalAdminRackId** all cluster nodes.

- InParam[0] = Rack ID

setClusterSystemLocalAdminRackPosition

Executes **setSystemLocalAdminRackPosition** all cluster nodes.

- InParam[0] = Rack position

setClusterSystemSyslogServerName

Executes **setSystemSyslogServerName** all cluster nodes.

- InParam[0] = Syslog server name

setClusterSystemUidEmailAddress

Executes **setSystemUidEmailAddress** all cluster nodes.

- InParam[0] = UID
- InParam[1] = email address

setClusterSystemUidEmailAddress

Executes **setSystemUidEmailAddress** all cluster nodes.

- InParam[0] = UID
- InParam[1] = email address

Disaster Recovery

Disaster can occur if the NAS head, storage array, or multiple devices are destroyed, or sustain critical hardware failures. The following text commands are provided to implement the recovery process. For detailed information on performing Disaster Recovery tasks, see the *HP NAS 8000 Users Guide*.

At a minimum, the recovery process requires a working NAS head with access to the DRF. The recovery process attempts to restore the NAS head and the storage array based on the contents of the DRF. In the case of a NAS head failure alone, no user data needs to be restored to the storage array. In the case of a storage array failure, the recovery process attempts to format the replacement array(s) to have the same LUN and volume group configuration as the current NAS registry indicates. After the storage array is recovered, user data can then be restored from backup tape. If both the head and storage array(s) fail, then the process used is to recover the head first, then recover the storage array(s).

recoverSystemStorageConfig

Recovers the NAS 8000 storage configuration based on the storage configuration contained in the NAS 8000 registry.

recoverSystemHeadFromDisasterRecoveryFile

Recovers the NAS 8000 head configuration based on the contents of the supplied Disaster Recovery file.

- inParam[0] = Disaster Recovery file name

getStorageVolumeDisasterRecovery

Displays the name of the Disaster Recovery volume.

- outParam[0] = Disaster Recovery volume name

getSystemDisasterRecoveryFileList

Displays the list of available Disaster Recovery files.

- outParam[0:N] = Disaster Recovery file list

recoverSystemHeadFromDisasterRecoveryFile

Recovers the NAS 8000 head configuration based on the contents of the supplied Disaster Recovery file.

- inParam[0] = Disaster Recovery file name

generateSystemDisasterRecoveryFile

Generates a new Disaster Recovery file based on the current configuration of the NAS 8000 head.

Boot Partition Settings

Use the following commands when you install a new version of the NAS OS or perform disaster recovery operations.

toggleSystemBootPartitions

Makes the active partition the one currently inactive, and vice-versa.

getSystemBootPartitionActive

Returns the name of the active boot partition.

- inParam[0] = Boot partition name
- outParam[0] = Partition active

setSystemBootPartitionActive

Sets the specified partition as the active boot partition.

- inParam[0] = Boot partition name
- inParam[1] = Partition active

getSystemBootPartitionList

Returns a list of boot partition names.

- outParam[0:N] = Partition names

getSystemBootPartitionOsBuildnumber

Returns the build number of the operating system

- inParam[0] = Boot partition name
- outParam[0] = Build number

getSystemBootPartitionOsVersion

Returns the version of the OS installed on the specified boot partition.

- inParam[0] = Boot partition name
- outParam[0] = OS version

index

T

U

W